

FIG. 1

→ DATA TRANSFERRED  
 ----> DATA NOT TRANSFERRED

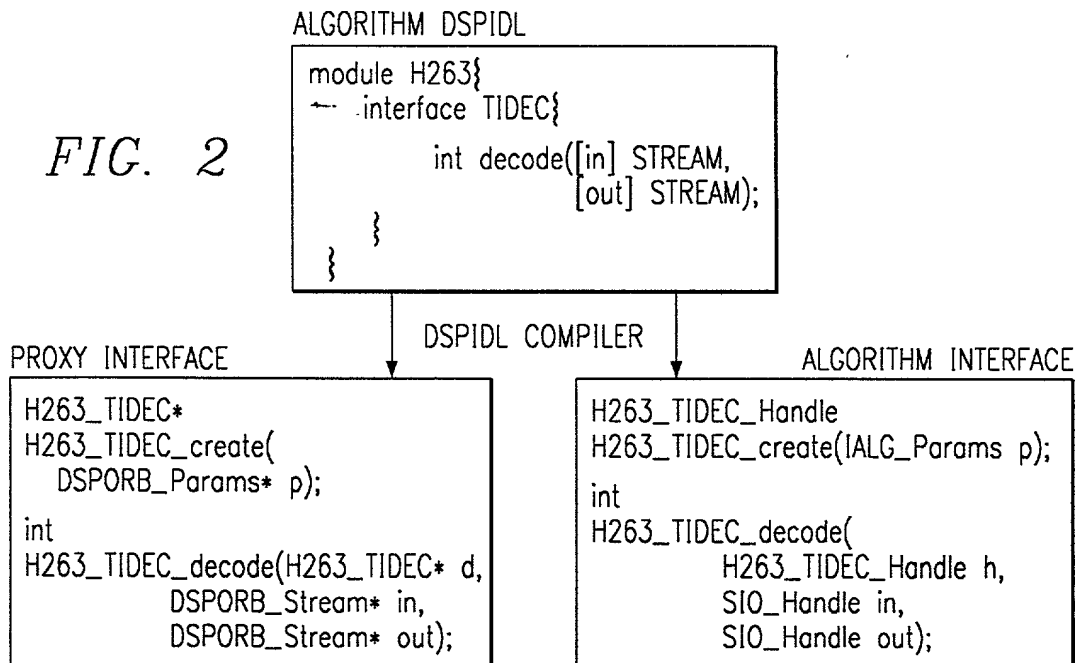
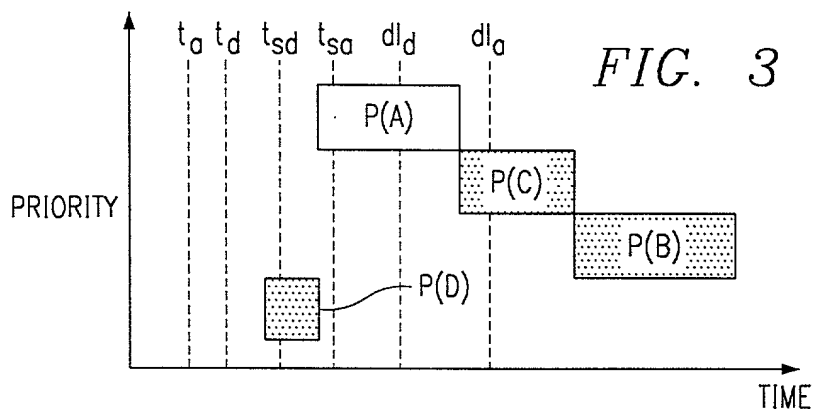
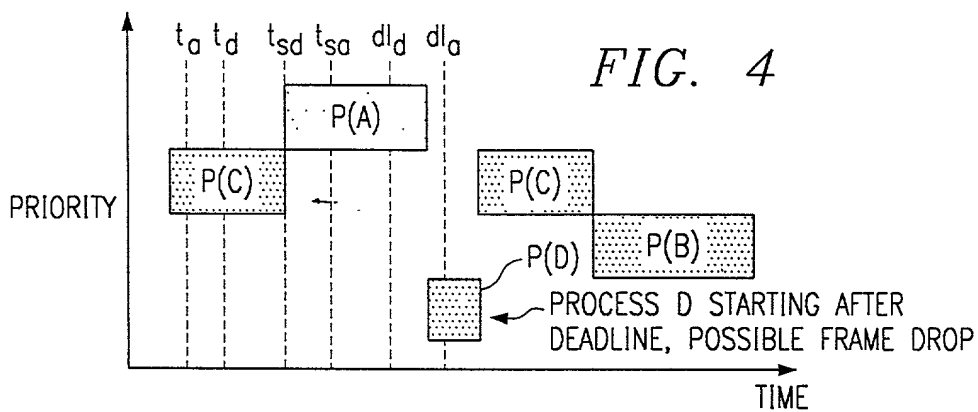


FIG. 2



$t_{sa}$  = LAST POSSIBLE TIME FOR PROCESS A  
TO START AND STILL MAKES ITS DEADLINE

$t_{sd}$  = LAST POSSIBLE TIME FOR PROCESS D  
TO START AND STILL MAKE ITS DEADLINE



$t_{sa}$  = LAST POSSIBLE TIME FOR PROCESS A  
TO START AND STILL MAKES ITS DEADLINE

$t_{sd}$  = LAST POSSIBLE TIME FOR PROCESS D  
TO START AND STILL MAKE ITS DEADLINE

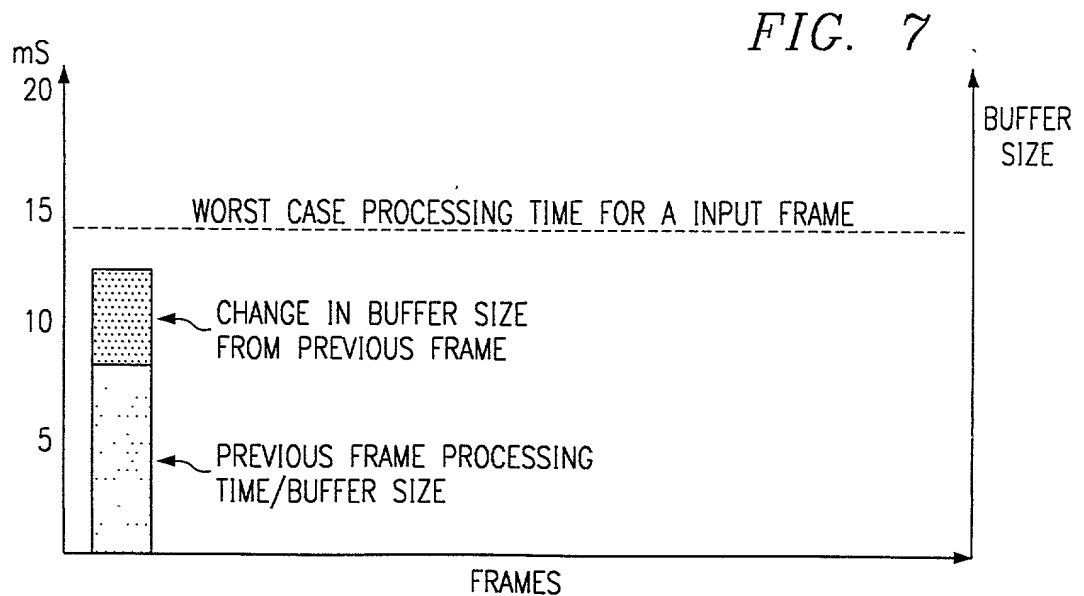
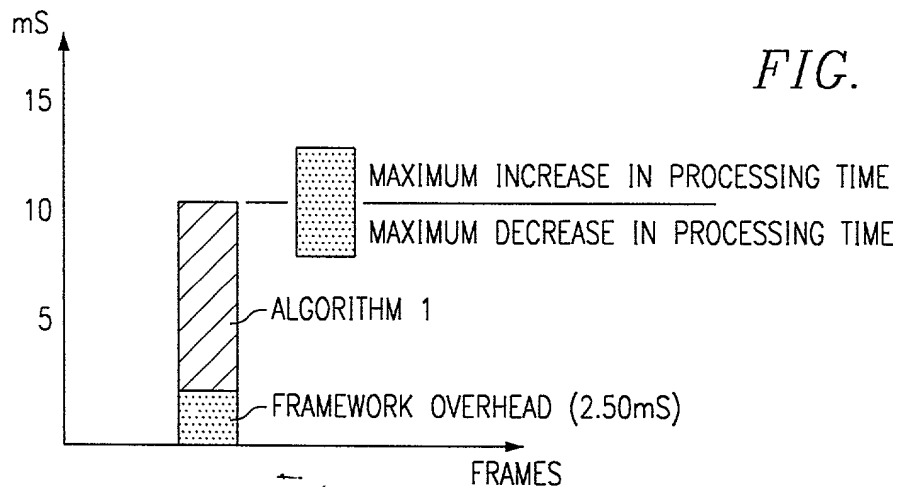
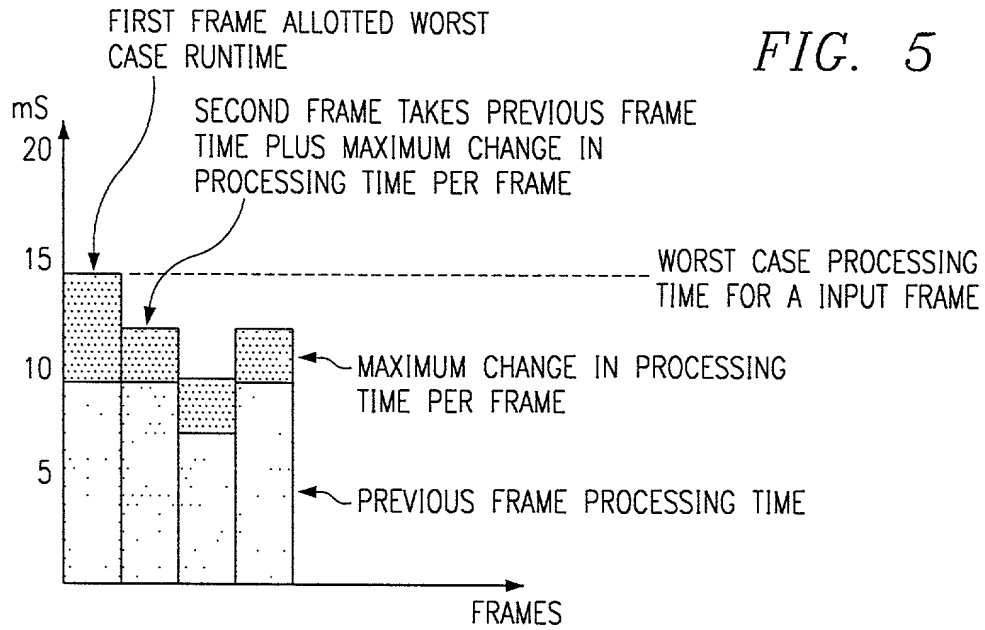


FIG. 8

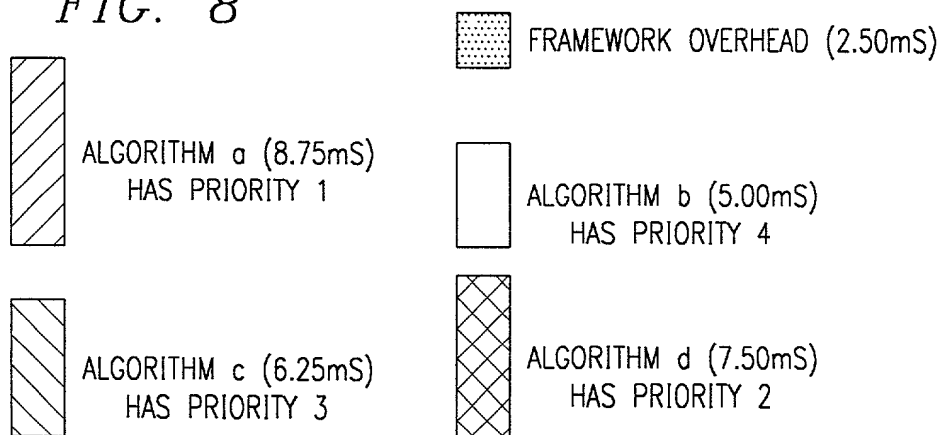
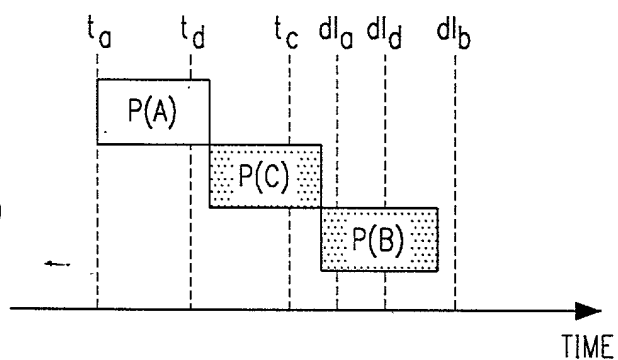
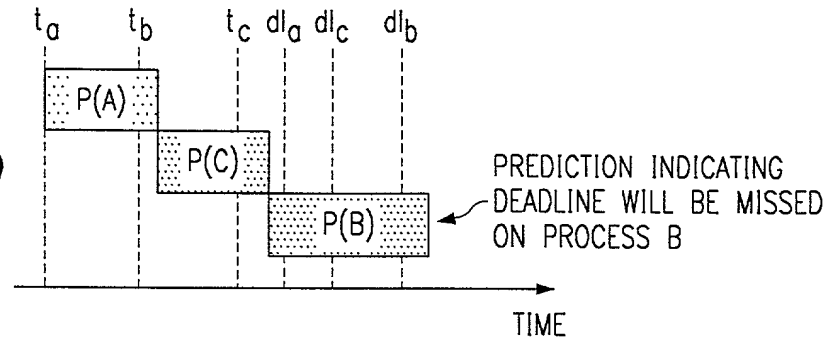


FIG. 9

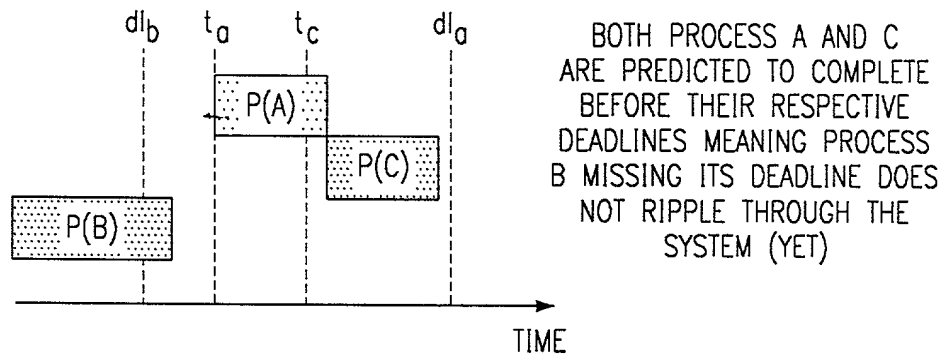


- $t_i$  = TIME STAMP ARRIVAL OF EACH DATA FRAME FOR THE RESPECTIVE PROCESS  
 $dl_i$  = DEADLINE FOR FINISHING PROCESSING OF EACH RECEIVED DATA FRAME  
 $P()$  = PREDICTION OF PROCESSING TIME FOR EACH RECEIVED DATA FRAME

FIG. 10

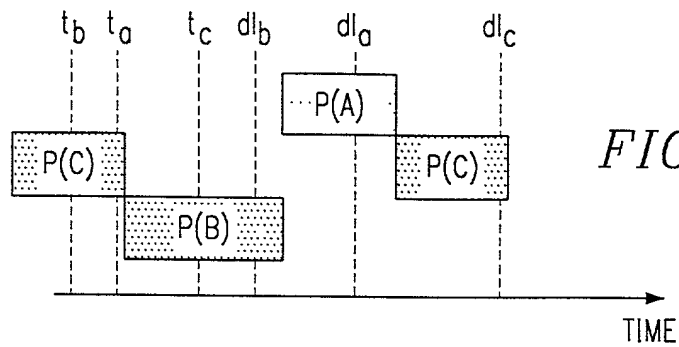


- $t_i$  = TIME STAMP ARRIVAL OF EACH DATA FRAME FOR THE RESPECTIVE PROCESS
- $dl_i$  = DEADLINE FOR FINISHING PROCESSING OF EACH RECEIVED DATA FRAME
- $P()$  = PREDICTION OF PROCESSING TIME FOR EACH RECEIVED DATA FRAME

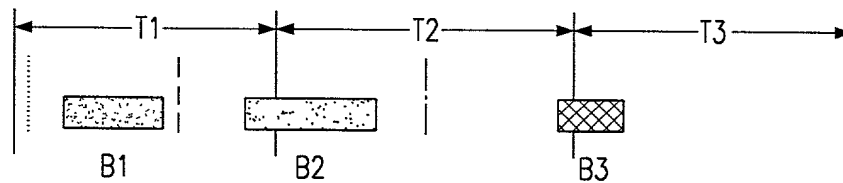
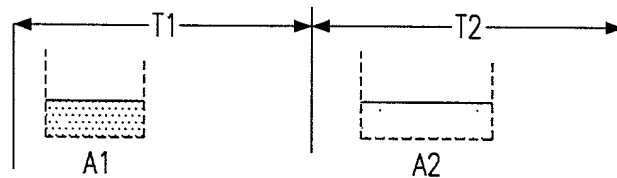


- $t_i$  = TIME STAMP ARRIVAL OF EACH DATA FRAME FOR THE RESPECTIVE PROCESS
- $dl_i$  = DEADLINE FOR FINISHING PROCESSING OF EACH RECEIVED DATA FRAME
- $P()$  = PREDICTION OF PROCESSING TIME FOR EACH RECEIVED DATA FRAME

FIG. 11



- $t_i$  = TIME STAMP ARRIVAL OF EACH DATA FRAME FOR THE RESPECTIVE PROCESS
- $dl_i$  = DEADLINE FOR FINISHING PROCESSING OF EACH RECEIVED DATA FRAME
- $P()$  = PREDICTION OF PROCESSING TIME FOR EACH RECEIVED DATA FRAME



- ..... ARRIVAL OF BUFFER B1
- ARRIVAL OF BUFFER B2
- ARRIVAL OF BUFFER B3

FIG. 14

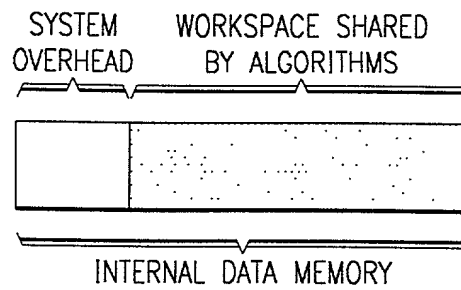


FIG. 15

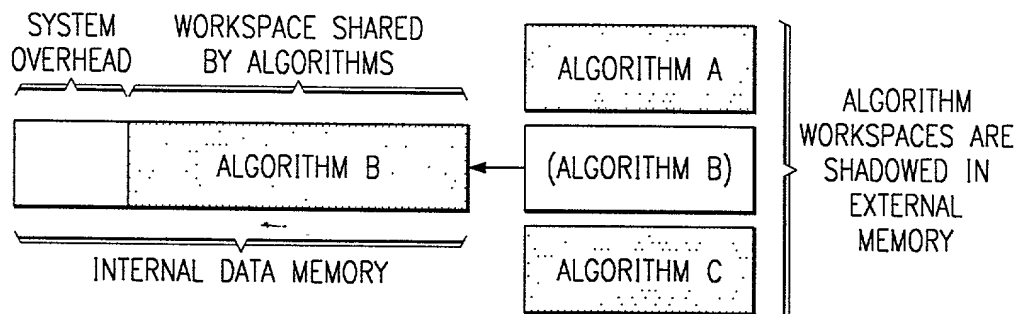
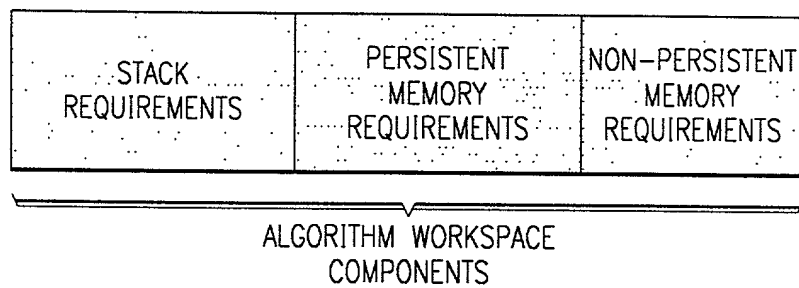
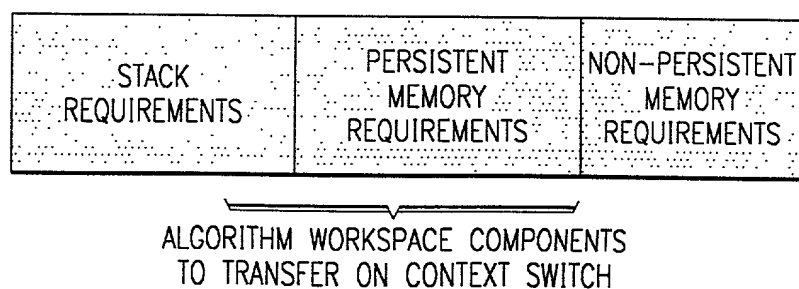
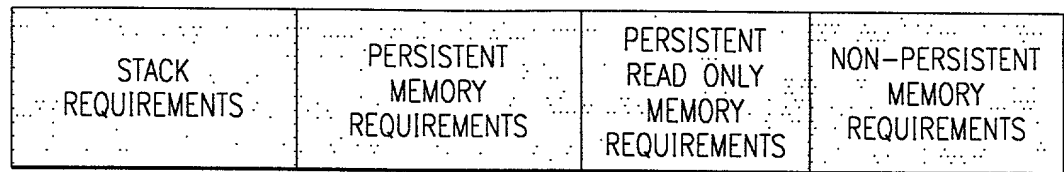


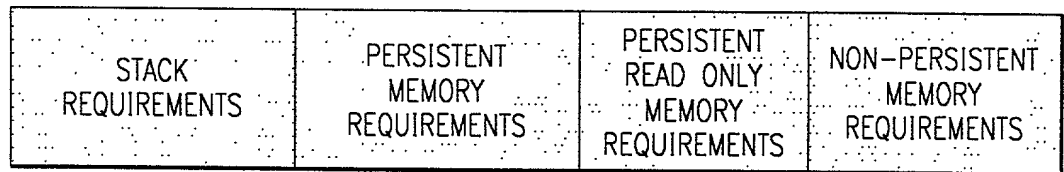
FIG. 16

FIG. 17



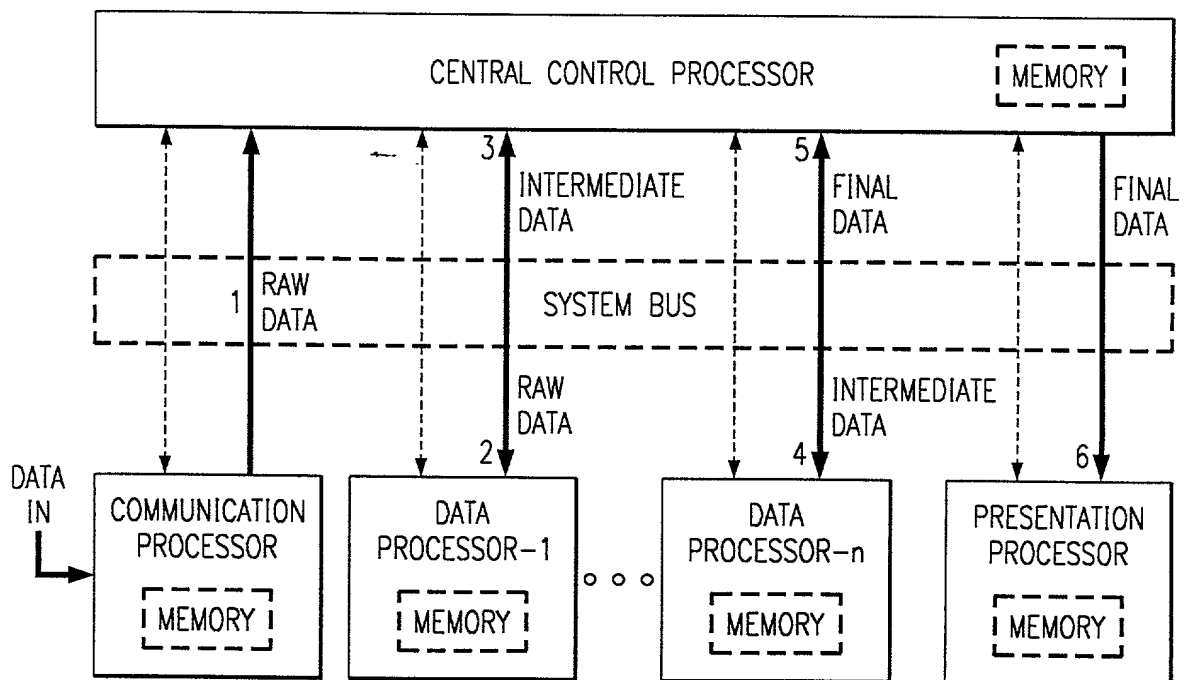


*FIG. 18* ALGORITHM WORKSPACE COMPONENTS TO  
TRANSFER IN PRIOR TO ALGORITHM EXECUTION  
IF ALGORITHM REQUIRES CONSTANT TABLES  
(CONTEXT SWITCH IN ONLY)



READ ONLY PERSISTENT MEMORY DOES  
NOT NEED TO BE TRANSFERRED OUT ON  
CONTEXT SWITCH. THEREFORE ALGORITHM  
PAGE CHANGE-OUT IS MORE EFFICIENT.

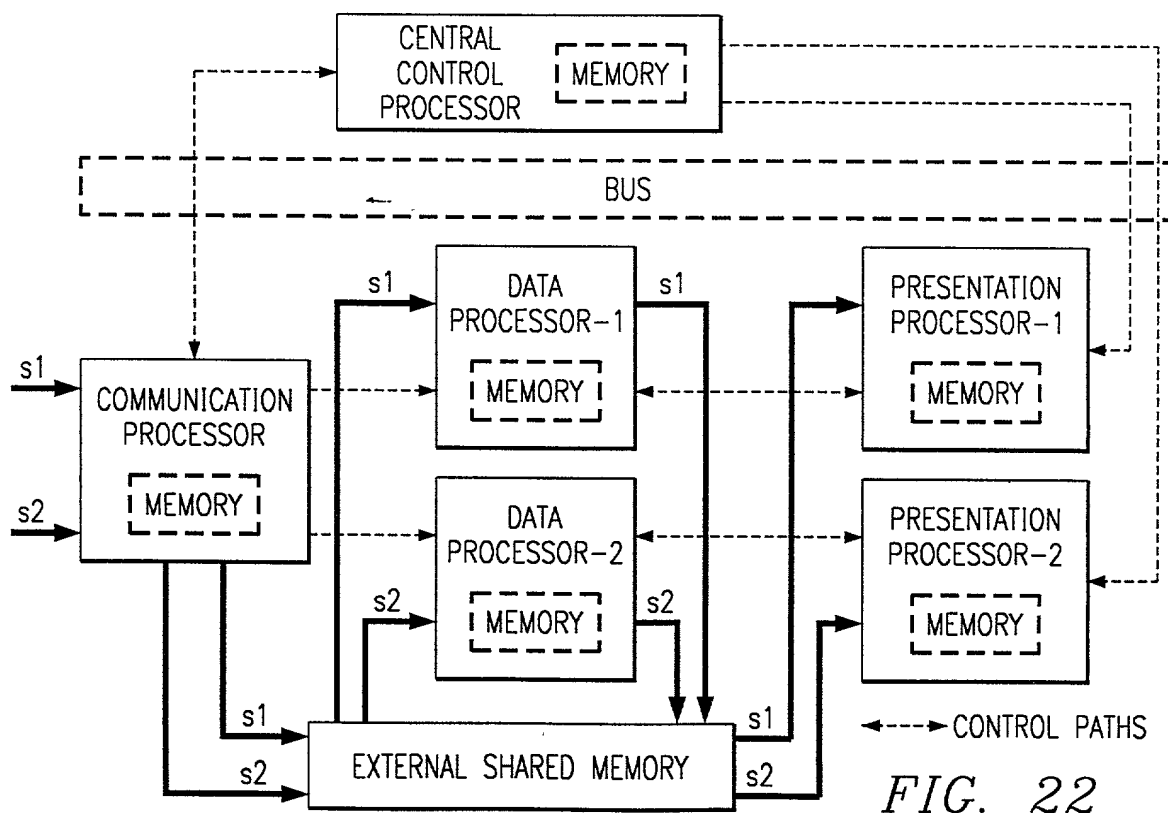
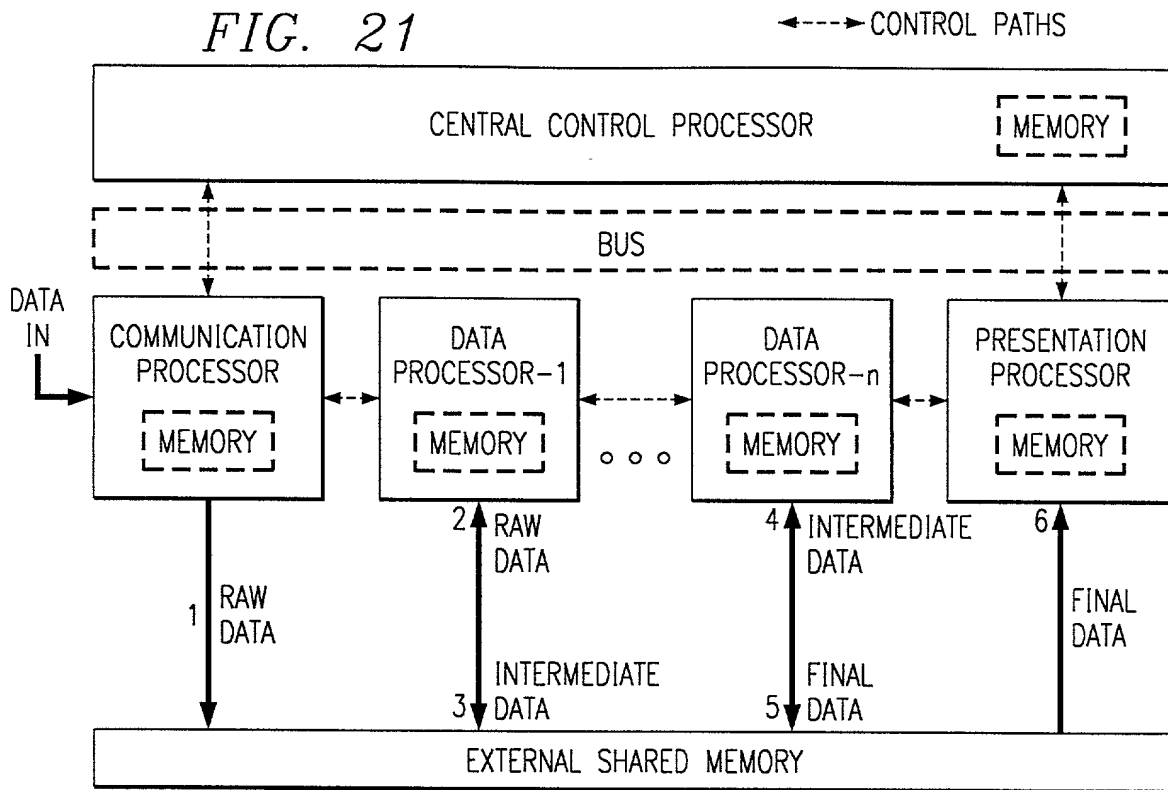
*FIG. 19*



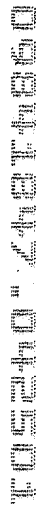
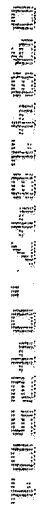
←---→ CONTROL PATHS

*FIG. 20*  
(PRIOR ART)





*FIG. 22*

[illegible][illegible]

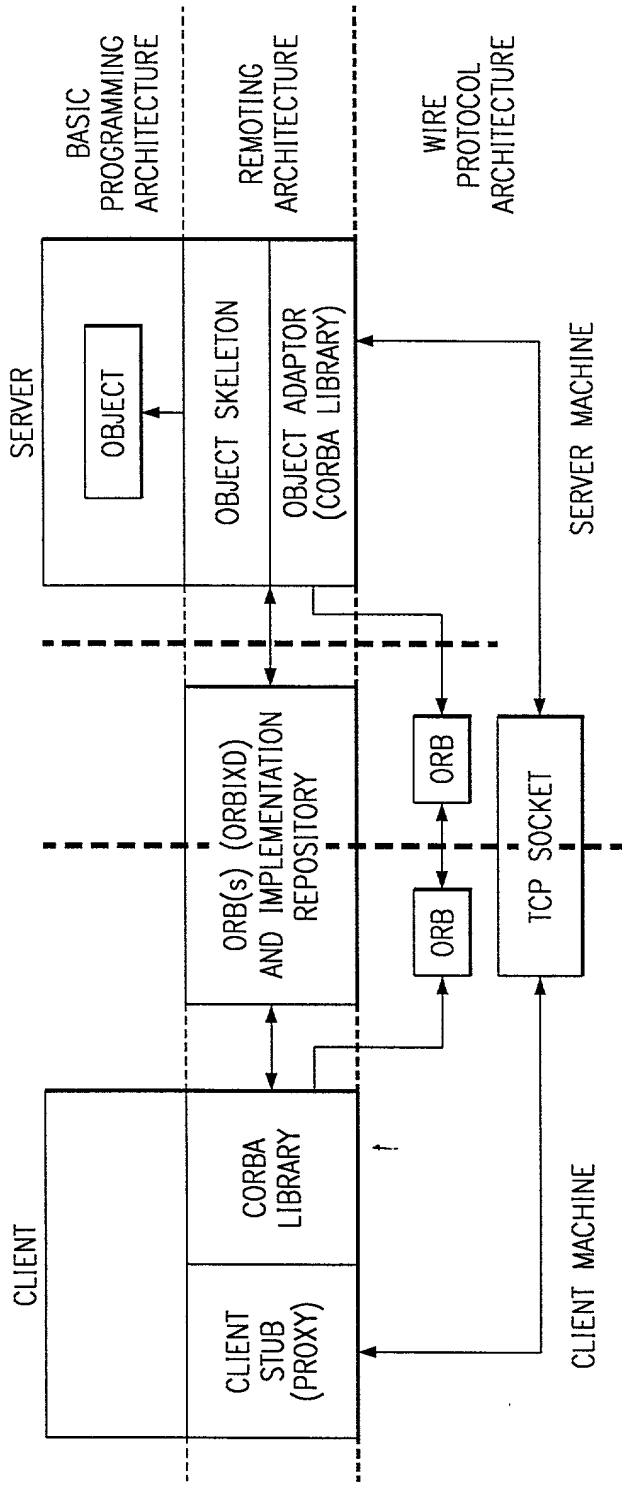


FIG. 25

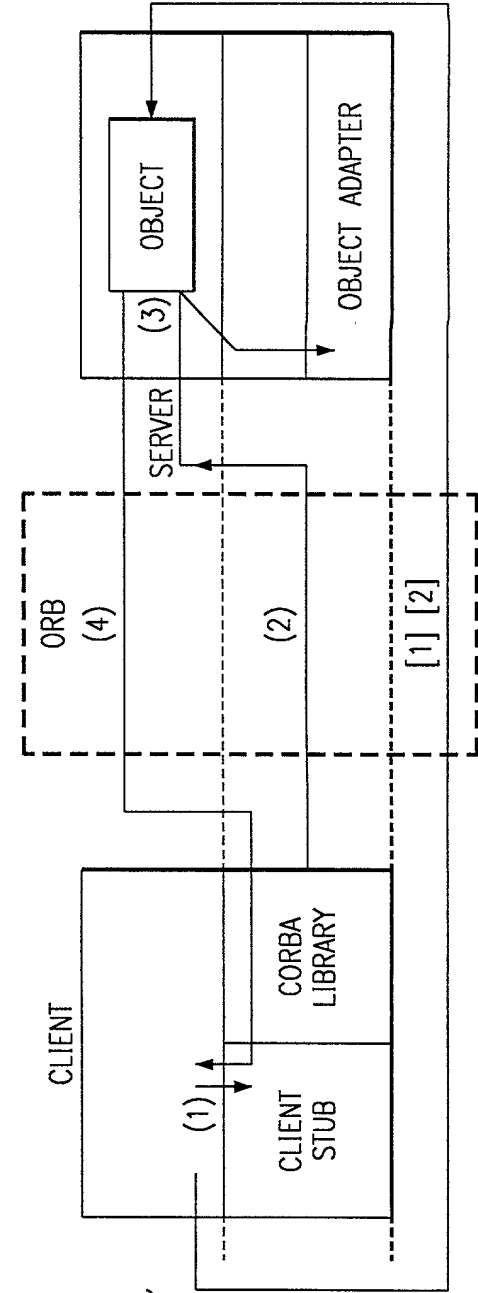


FIG. 26a

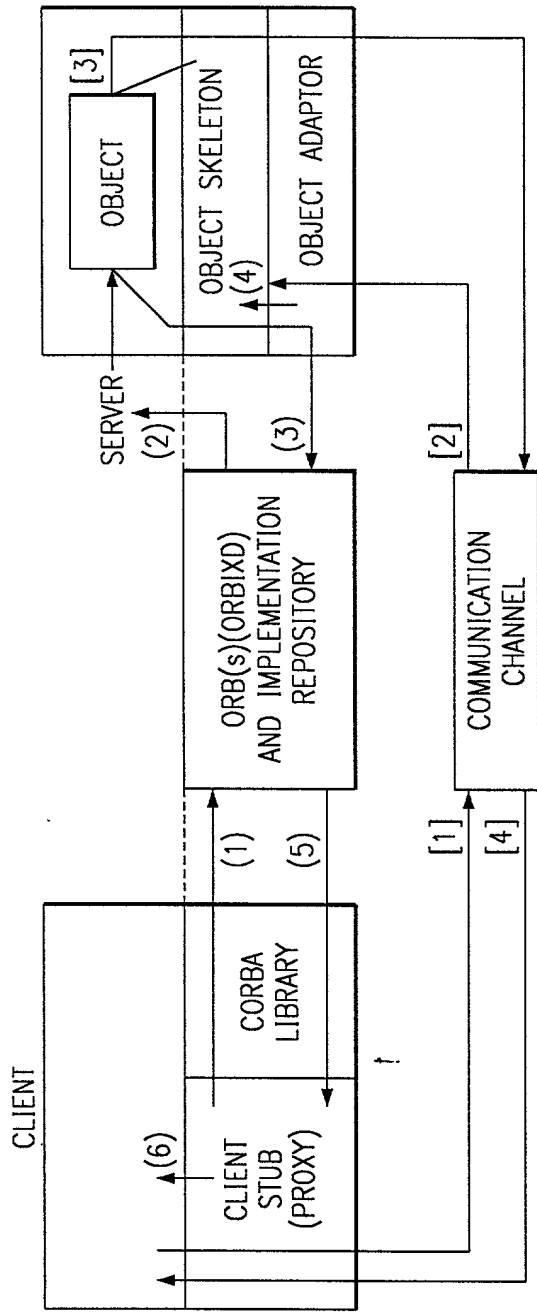


FIG. 26b

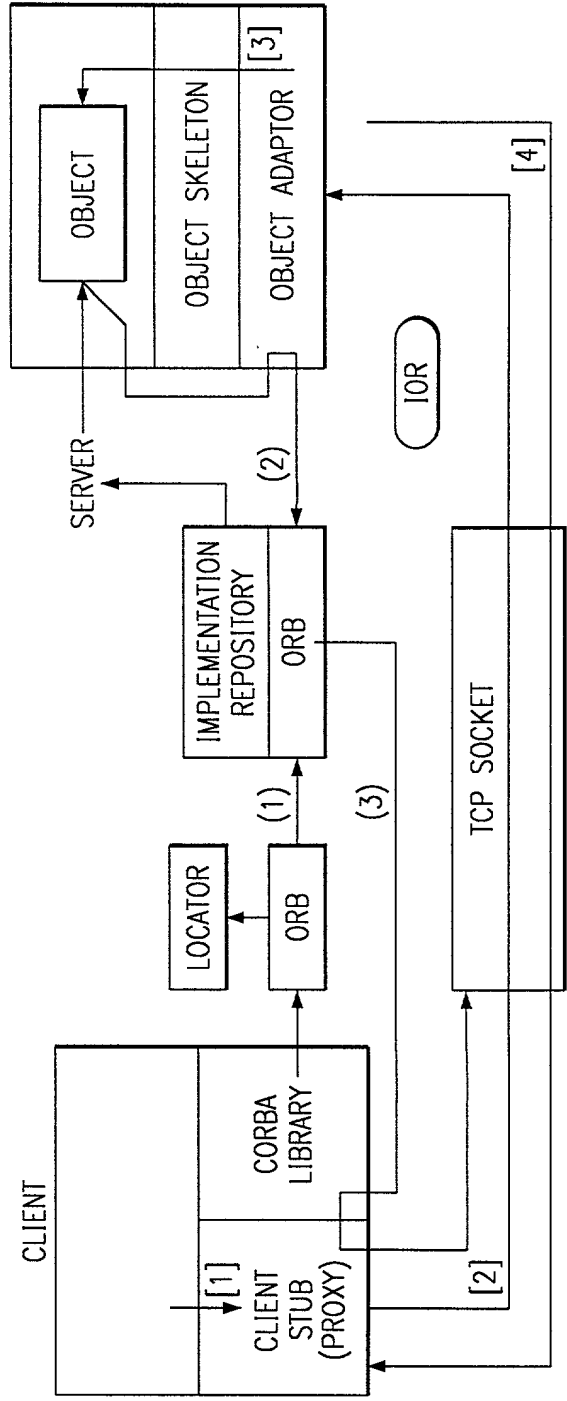


FIG. 26c

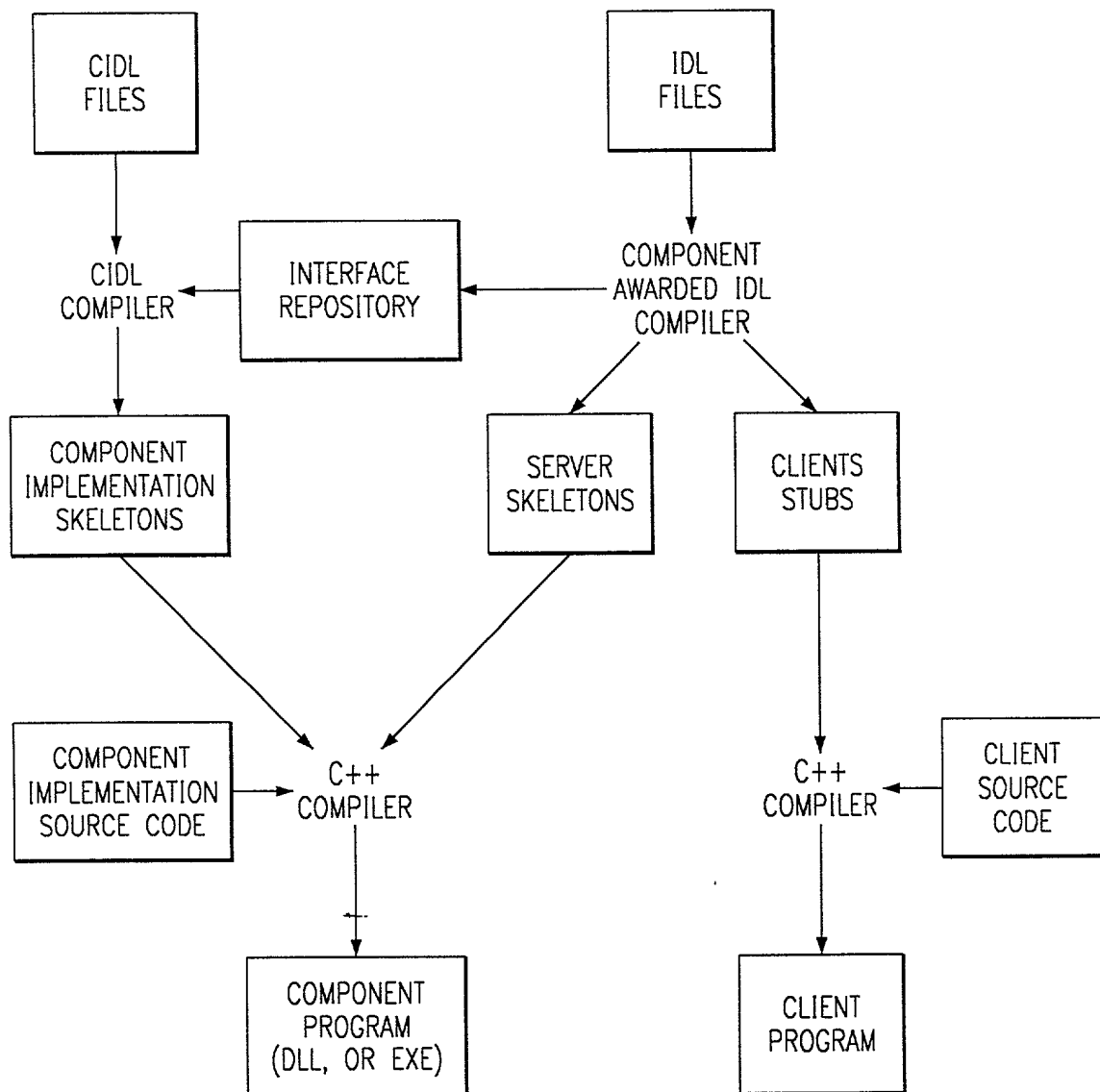


FIG. 27